NASA'S MISSION TO PLANET EARTH

EARTH PROBES

DATA INFORMATION SYSTEM

EOS

EARTH OBSERVING SYSTEM

Management Agent Services

Dennis Kavanagh

19 January 1995

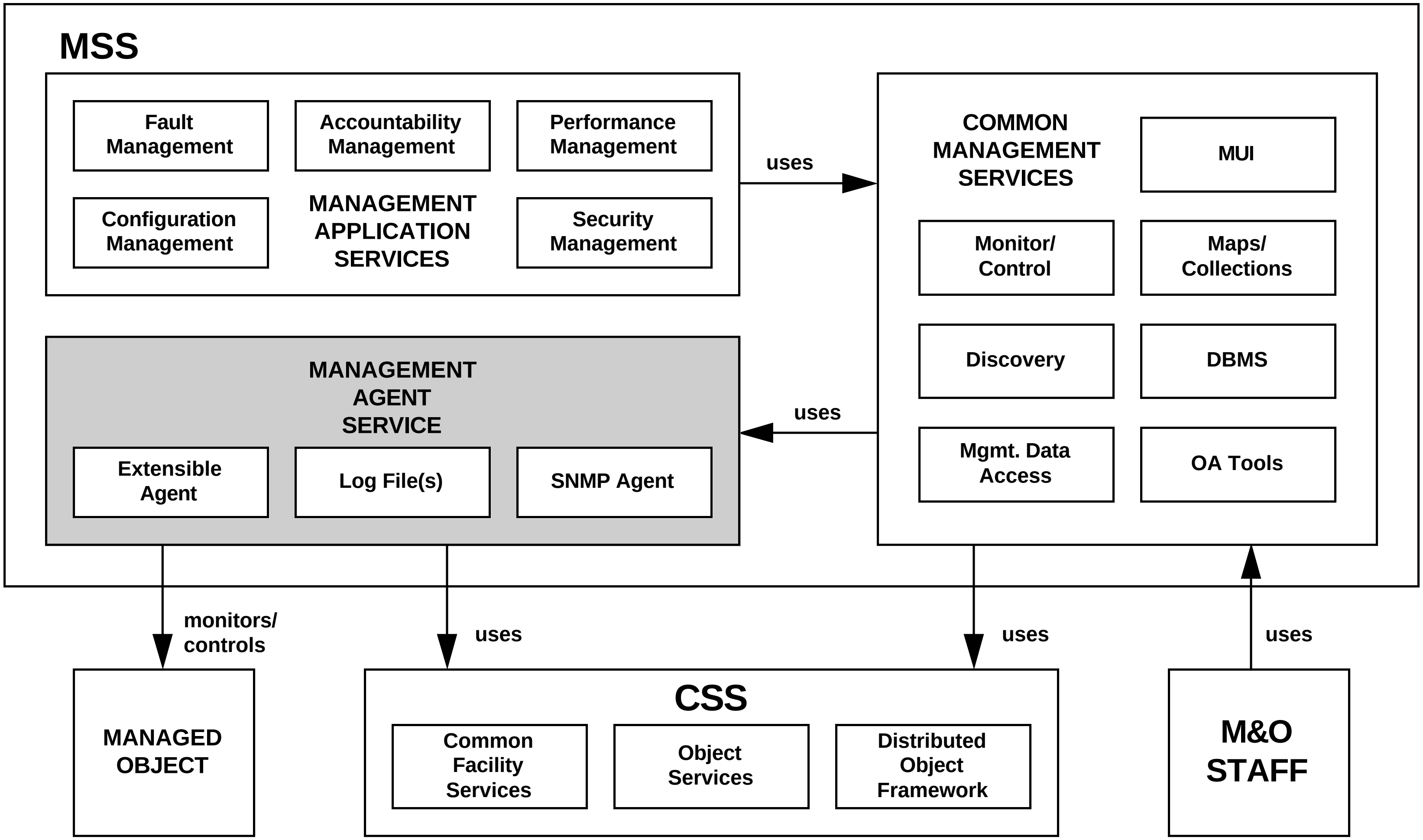
Management Agent Services

- Capabilities by Release
- Context Diagram
- Design Description
- Key Technologies
- Migration/Evolution
- Scenario

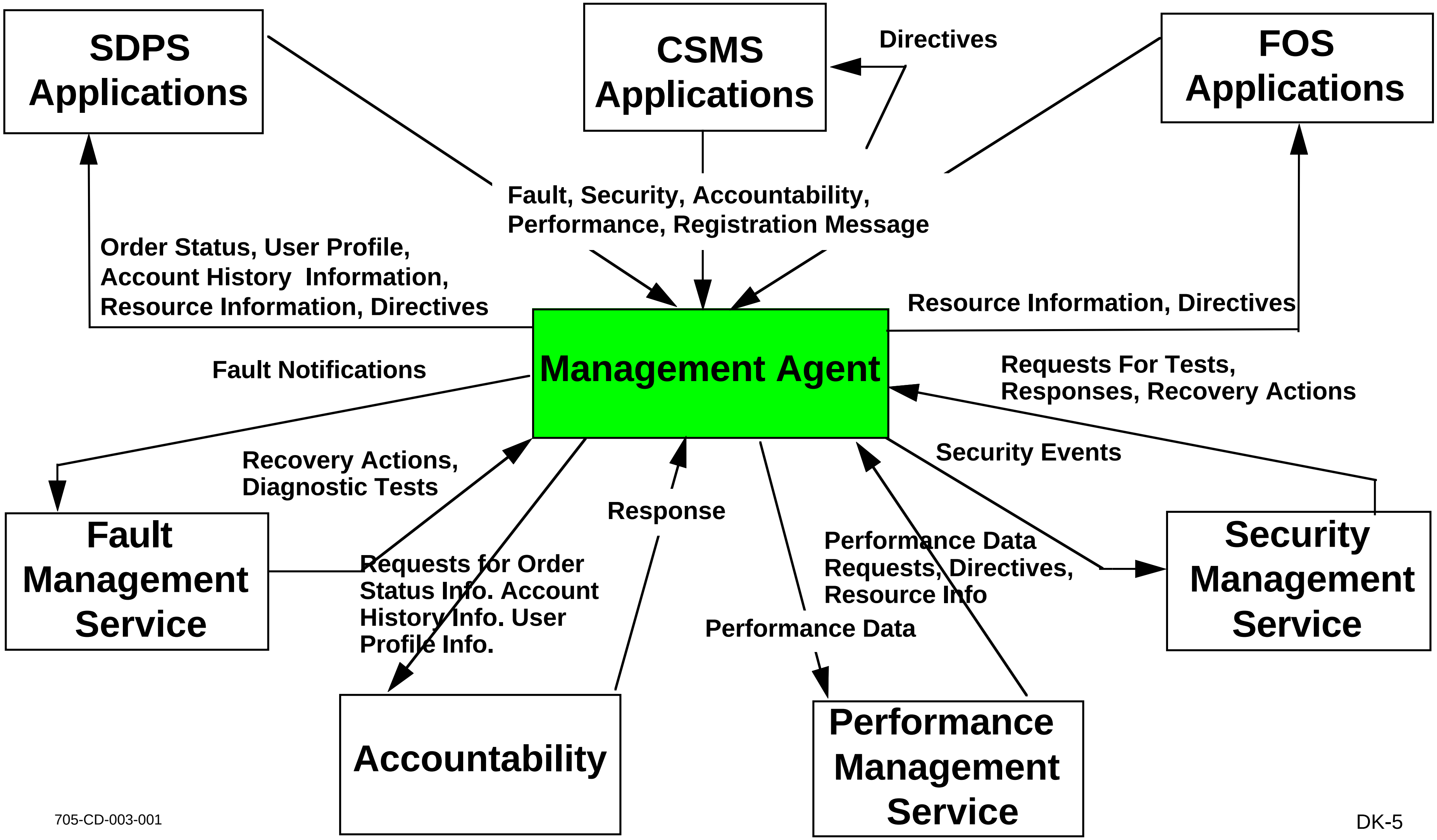
Management Agent Services Capabilities By Release

IR-1	Release-A
• Services - log/monitor applications/ host messages - provide extensible Agent/MIB architecture - forward trap events - provide SNMP v1 - provide MIB-I and MIB-II	• Services - log/monitor applications/host messages (enhanced) - provide extensible Agent/MIB architecture (enhanced) - *forward trap events* - *provide SNMP v1* - *provide MIB-I and MIB-II*

MSS Subsystem Design



Management Agent Services Context



SDPS Applications

CSMS Applications

Directives

FOS Applications

Fault, Security, Accountability, Performance, Registration Message

Order Status, User Profile, Account History Information, Resource Information, Directives

Resource Information, Directives

Management Agent

Fault Notifications

Requests For Tests, Responses, Recovery Actions

Recovery Actions, Diagnostic Tests

Security Events

Response

Fault Management Service

Requests for Order Status Info. Account History Info. User Profile Info.

Performance Data Requests, Directives, Resource Info

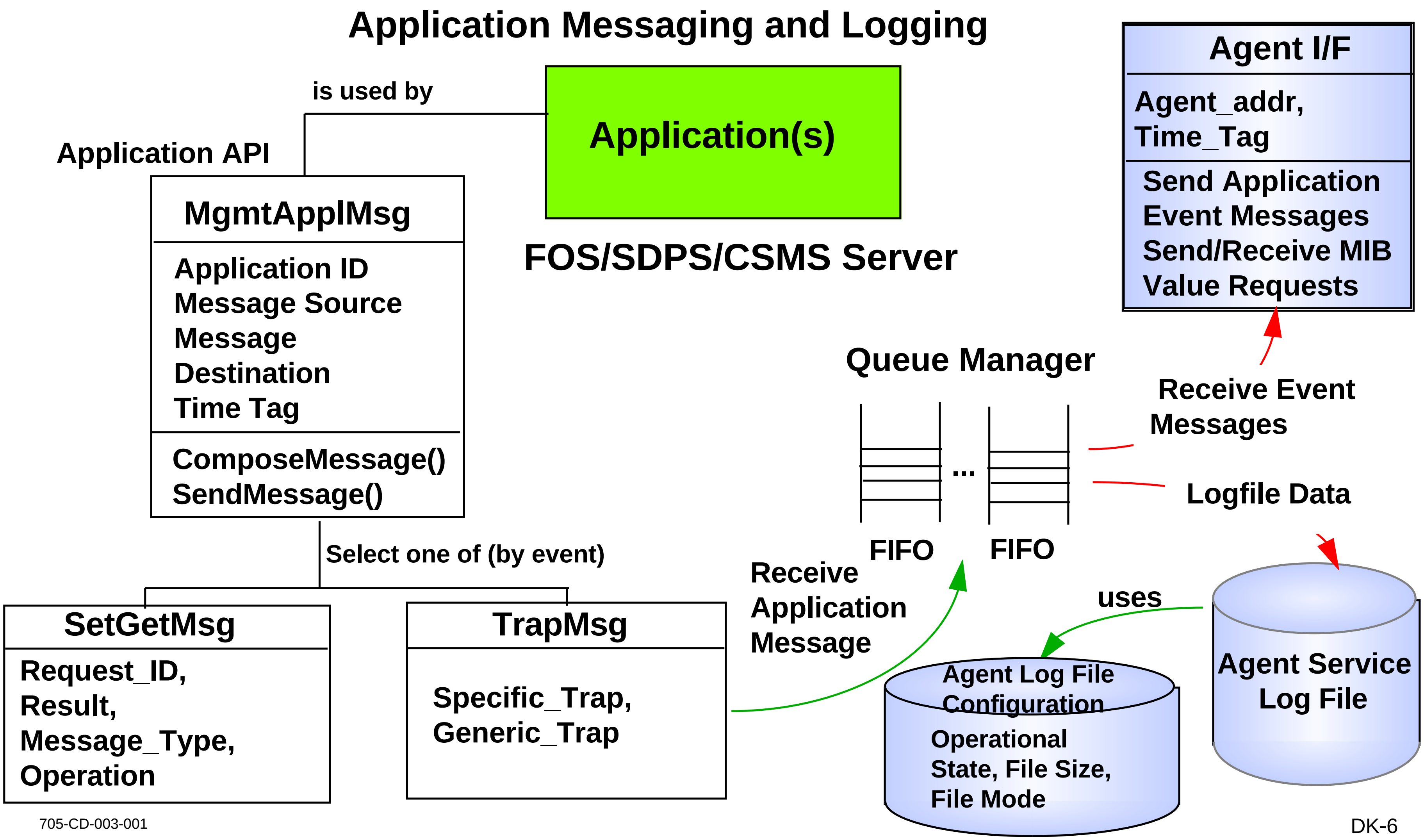
Performance Data

Security Management Service

Accountability

Performance Management Service

Management Agent Service Design Decomposition (part 1)



Application Messaging and Logging

Application API

is used by

Application(s)

FOS/SDPS/CSMS Server

MgmtApplMsg

Application ID
Message Source
Message
Destination
Time Tag

ComposeMessage()
SendMessage()

Select one of (by event)

SetGetMsg

Request_ID,
Result,
Message_Type,
Operation

TrapMsg

Specific_Trap,
Generic_Trap

Agent I/F

Agent_addr,
Time_Tag

Send Application
Event Messages
Send/Receive MIB
Value Requests

Queue Manager

...

FIFO

FIFO

Receive Event
Messages

Logfile Data

Receive
Application
Message

uses

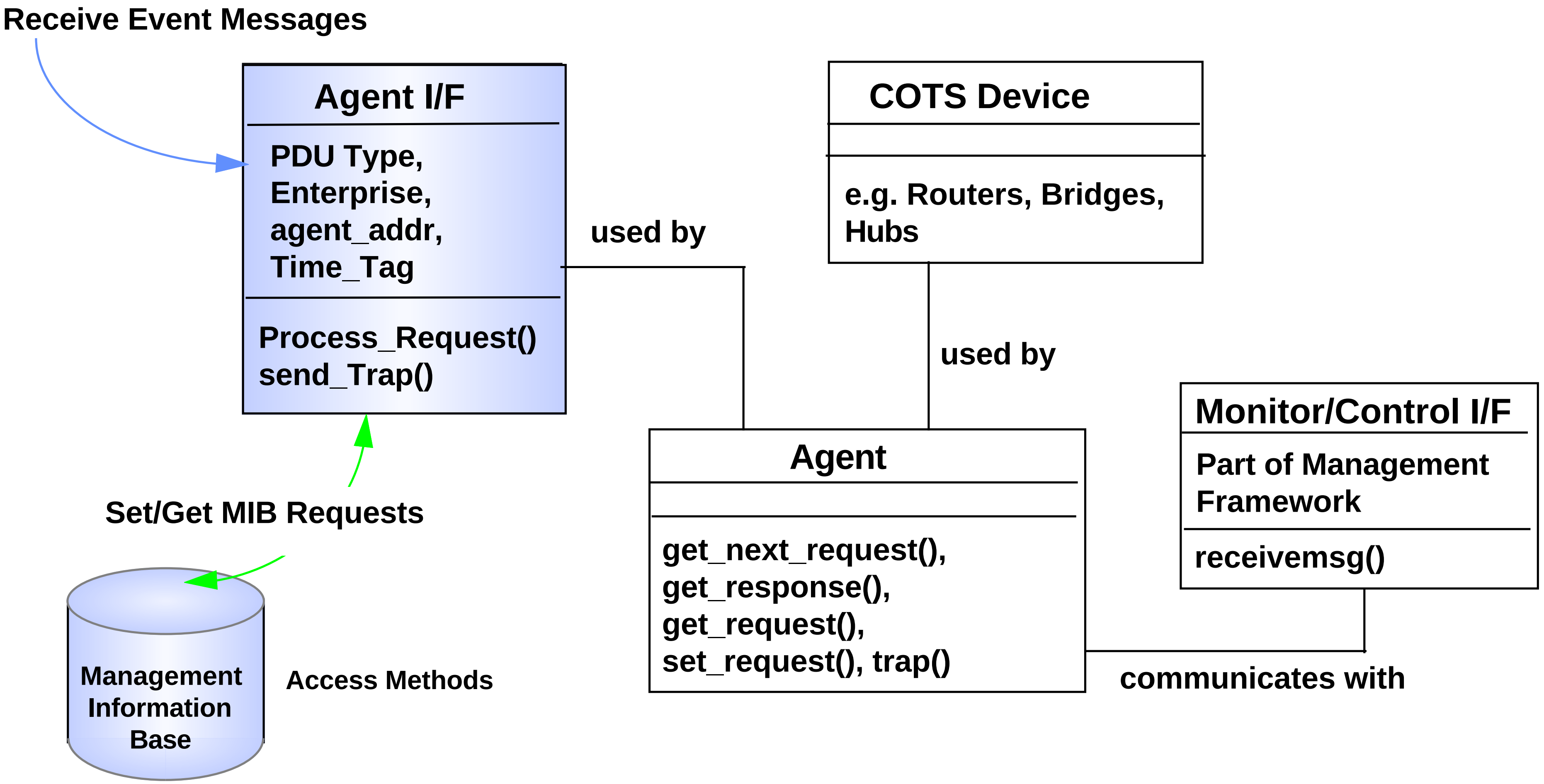
Agent Log File
Configuration

Operational
State, File Size,
File Mode

Agent Service
Log File

Management Agent Service Design Decomposition (part 2)

Provide Event Message Request/Response



Management Agent Services Summary

Key Technology Selection

Agent Analysis found

- SNMP Agent(s) and Extensible Agent(s) are sufficient
- RMON is not needed in Release A; expect to add in Release B

SNMP vs CMIP

- OSI/CMIP standard is still a draft standard; it is not commercially widespread
- HP OpenView supports SNMP and draft of CMIP

Migration and Evolution

HP OpenView is SNMPv1 Currently

- Migrating to SNMPv2 in next release of OpenView

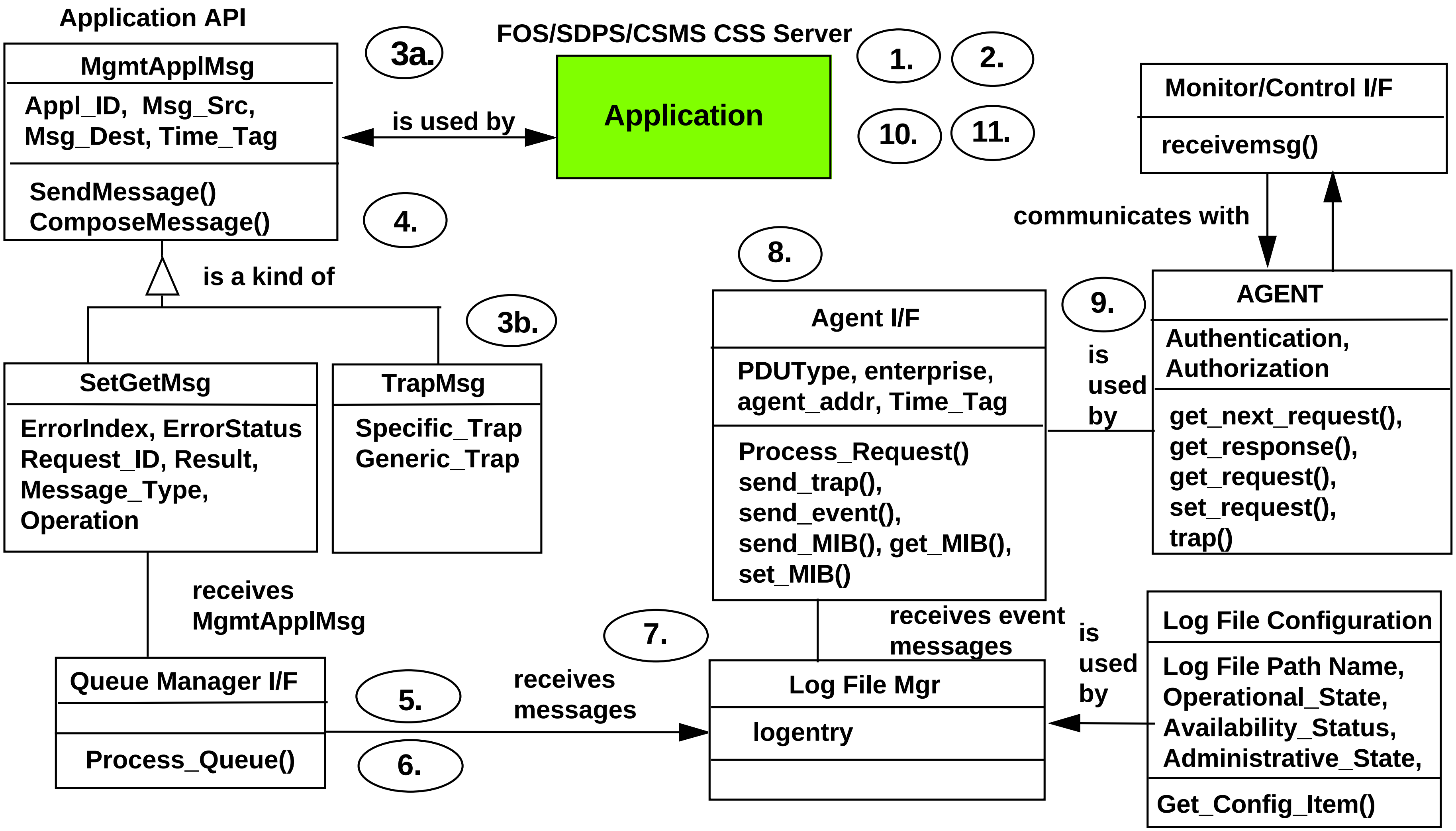
SNMP to OSI

- SNMPv2 provides OSI translation

SNMPv1 to SNMPv2

- Sub-Agent available for translation

Management Agent Services Scenario



Application API

MgmtApplMsg

Appl_ID, Msg_Src, Msg_Dest, Time_Tag

SendMessage()
ComposeMessage()

3a.

is used by

FOS/SDPS/CSMS CSS Server

Application

1. 2. 10. 11.

Monitor/Control I/F

receivemsg()

communicates with

4.

is a kind of

8.

3b.

SetGetMsg

ErrorIndex, ErrorStatus
Request_ID, Result,
Message_Type,
Operation

TrapMsg

Specific_Trap
Generic_Trap

Agent I/F

PDUType, enterprise,
agent_addr, Time_Tag

Process_Request()
send_trap(),
send_event(),
send_MIB(), get_MIB(),
set_MIB()

9.

is used by

AGENT

Authentication,
Authorization

get_next_request(),
get_response(),
get_request(),
set_request(),
trap()

receives MgmtApplMsg

7.

receives event messages

is used by

Queue Manager I/F

Process_Queue()

5.

6.

receives messages

Log File Mgr

logentry

Log File Configuration

Log File Path Name,
Operational_State,
Availability_Status,
Administrative_State,

Get_Config_Item()

Management Agent Service Scenario

1. Application registers with MSS system through MSS provided API

   ComposeMessage (AppID=application1, Msg_Dest = Reg_File,Time_Tag = 121224, MessageType = Register)
   SendMessage (Message_Record)

2. Application detects a file access error

3. Application error handler uses the MSS provided API to compose Trap Event Message using TrapMsg form of ComposeMessage()

   ComposeMessage (AppID = application1, Msg_Dest = Log_File, Time_Tag = 121224, MessageType = Fatal, Operation=Log, generic_trap = 6, specific_trap = 13)

4. Application uses the MSS provided API to send message record to Management Agent Services

   SendMessage (Message_Record, Severity_Level = Fatal)

5. CSS Event API decodes Severity Level and puts message record on Agents Service Trap Events queue and notifies Queue_Mgr

Management Agent Service Scenario

6. Queue Mgr. identifies the Trap message record on Fatal Events queue

7. Queue Mgr. forwards the Trap Event message record to LogMgr

```
SendMessage (  Message_Record, Severity_Level = Fatal )
```

8. Queue Mgr. forwards message record using Agent API

```
send_trap ( Message_Record, PDUType=trap,  agent_addr=192.10.9.23, Time_Tag=121229 )
```

9. Agent API verifies authenticity/authorization and encodes message as SNMP Trap and sends to enterprise manager.

```
trap ( Trap_Message_Record )
```

10. Application unregisters with MSS System

```
ComposeMessage (AppID=application1, Msg_Dest = Reg_File,Time_Tag = 121224, MessageType = UnRegister)
SendMessage (  Message_Record )
```

11. Application takes appropriate step(s)